QVS: Quality-aware Voice Streaming for Wireless Sensor Networks

Liqun Li\(^1\), Guoliang Xing\(^2\), Limin Sun\(^1\)

\(^1\) Institute of Software, Chinese Academy of Sciences, China
\(^2\) Department of Computer Science and Engineering, Michigan State University, USA

Abstract

Recent years have witnessed the pilot deployments of audio or low-rate video wireless sensor networks for a class of mission-critical applications including search and rescue, security surveillance and disaster management. In this paper, we report the design and implementation of Quality-aware Voice Streaming (QVS) for wireless sensor networks. QVS is built upon SenEar, a new sensor hardware platform we developed for high-bandwidth wireless audio communication. QVS comprises several novel components, which include an empirical model for online voice quality evaluation and control, dynamic voice compression/duplication adaptation for lossy wireless links, and distributed stream admission control that exploits network capacity for rate allocation. We have extensively tested QVS on a 20-node network deployment. Our experimental results show that QVS delivers satisfactory voice quality under a range of realistic settings while achieving high network capacity utilization.

1 Introduction

Recent years have witnessed the deployments of wireless sensor networks in a class of mission-critical applications including coal mine monitoring [15], security surveillance [12] and disaster management. In contrast to the traditional wireless sensor networks that focus on low duty-cycle and long-lived data collection, audio or low-rate video sensor networks are often more desired for these applications due to their capability of supporting short-term, real-time and high-rate data streaming. However, wired network configuration is often not an option for these applications due to high installation/cabling cost or constraints of deployment environments. For instance, audio sensor networks can be deployed in buildings and assist post-earthquake search and rescue operations. In such a scenario, sensor networks must be able to function even when the prevalent wired communication and power infrastructure is destroyed.

In this paper, we address the problem of supporting voice streaming over wireless sensor networks (WSNs).
QVS has been extensively tested on a 20-node network deployment. Our experimental results show that it delivers satisfactory voice quality under a range of realistic settings while achieving high network capacity utilization. QVS is particularly suitable for short-term high-quality voice transfers in emergency situations such as post-earthquake search and rescue.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3, we provide an overview of system architecture. We discuss voice quality modeling in Section 4. The compression/duplication adaptation and admission control are presented in Section 5 and Section 6, respectively. We offer experimental results in Section 7 and conclude the paper in Section 8.

2 Related Work

Voice transfer over low-power multi-hop wireless networks has been investigated in several recent studies. Mangharam et al. [15] developed a system for transferring voice over sensor networks based on the FireFly nodes. The nodes in the system operate in a global TDMA schedule. A specialized radio is used to achieve hardware-based time synchronization. However, TDMA often incurs a high overhead and performs poorly in response to network dynamics such as variation of traffic load and link/topology quality. The design goal of the system in [15] is to support the transfer of one (two-way) voice stream. In contrast, our system is designed to transfer a number of voice streams in dynamic environments subject to voice quality and network capacity constraints. Moreover, the quality of voice stream that is transferred in the network is not monitored in [15]. In contrast, our design integrates a voice quality model and ensures the quality of voice streams by several mechanisms including dynamic compression and duplication adjustment and quality-aware admission control.

In [7], several performance metrics of voice streaming over Zigbee networks are studied, which include throughput, latency, packet loss and jitter. Luo et al. developed a sensor network system for cooperative audio storage and retrieval [14]. However, real-time audio streaming is not supported by the system. Several real-time communication protocols have been developed for WSNs. The SPEED [9] and RAP [13] protocols are designed to meet the end-to-end deadlines of flows based on the velocity of packets. In [5], the transmission power of nodes is adjusted based on the slack time before the delivery deadline of a packet. Real-time guarantees are achieved in SUPPORTS [11] by queue management and packet scheduling. In [1], capacity bounds for real-time transmissions in sensor network are derived. However, these studies do not address the issue of transferring voice streams with quality guarantees. In particular, our experiments show that the reliability of a multi-hop network path has a much more significant impact than end-to-end delay on the quality of voice streams.

Congestion control in multi-hop wireless network has been recently studied. Several solutions (CODA [22], Fusion [10], and WCPCap [17]) can detect congestion and decrease the data rates of contending links to achieve the fair sharing of channel capacity. However, congestion control protocols are not suitable for transmitting voice streams due to several reasons. First, fair sharing of channel capacity will lead to low voice quality when the number of voice streams is large. Moreover, adaptive rate control mechanisms such as Additive Increase Multiplicative Decrease (AIMD) is often employed by source nodes to achieve high utilization of bandwidth. However, it is shown [19] that such mechanisms take a long time for data rate to converge in low-rate wireless links, which would cause significant variation in voice quality.

Recently, a rate allocation scheme is proposed in [19] to achieve fairness based on channel saturation throughput and receiver capacity. We adopt the similar concepts in allocating data rates for voice streams in this work. However, different from [19], our rate allocation scheme is based on a novel admission control mechanism that accounts for the quality of voice streams.

3 System Architecture

The architecture of our system is shown in Fig. 1. We first briefly discuss the hardware platform and protocol stack in Section 3.1. We then provide an overview of QVS (whose components are shaded in Fig. 1) in Section 3.2.

![Figure 1. The architecture of QVS voice streaming system.](image)

3.1 SenEar Sensor Network Platform

We have developed a low-cost hardware platform called SenEar (Fig. 2). SenEar consists of a communication/CPU board and a sensor board. To support high-bandwidth voice processing, SenEar uses a 32-bit At-
mel AT91SAM7S256 microcontroller [2] with 256 KB of flash memory and 64 KB of RAM. The transceiver is Chipcon CC1100 [21] which has a 64-byte FIFO buffer and the maximum data rate is 500 Kbps. The CC1100 supports a hardware-based low-power-listening implementation, Wake-on-Radio, which enables the CPU to remain in sleep while the radio operates in a duty cycle.

An electret condenser microphone (ECM) is used for voice sampling. As the frequency of human voice ranges from 340 Hz to 3400 Hz, the sampling rate must be at least 6.8 KHz according to the Nyquist theory. We adopt a sampling rate of 8 KHz to ensure good voice quality. As a result, the minimum data rate of a voice stream without compression is 64 Kbps. To reduce the bandwidth usage, we implement a software Adaptive Differential Pulse Code Modulation (ADPCM) [8] to compress the sampled data. ADPCM is a simple compression algorithm suitable for platforms with limited computational power. The microphone outputs 8-bit values that are compressed to 5-bit, 4-bit, 3-bit or 2-bit ADPCM codes. The corresponding raw data rates are 40 Kbps, 32 Kbps, 24 Kbps and 16 Kbps, respectively. In order to reduce the noise in voice samples, a third-order band-pass filter is used to attenuate the frequencies outside of the range 340-3400 Hz.

The CC1100 radio has a current consumption of 28.9 mA in TX mode and 15.2 mA in RX mode. The current consumption of AT91SAM7S256 in full-speed mode is less than 60 mA. We measured the average current consumption of the whole system at full load to be about 70 mA. SenEar can be connected with replaceable external sockets for AA, C or D-cell batteries. When four AA, two C or D-cell batteries are used, the system with constant voice streaming will last about 81, 120, and 293 hours, respectively. SenEar is mainly designed for short-term emergency situations. Specifically, QVS comprises the following components that are designed to meet the above requirements: 1) an empirical voice model that quantifies the voice quality of a stream signed to meet the above requirements: 1) an empirical voice model that quantifies the voice quality of a stream based on transport parameters including packet loss ratio and delay. The model is used by the source/sink of a stream for automatic voice quality evaluation and control; 2) an adaptation mechanism that dynamically adjusts voice compression/duplication ratios to maintain desirable voice quality in face of dynamic link conditions; 3) a distributed admission control algorithm that assigns stream data rates based on the stream quality requirement as well as the available network capacity measured by each node locally.

Figure 2. The SenEar sensor network platform.
4 Modeling Voice Quality for Lossy Links

We now discuss an empirical voice quality model that is used by QVS to achieve online quality control for voice streaming. The model quantifies the voice quality of a stream based on transport parameters including packet loss ratio and delay. The quantification of voice quality is fed back to other components of QVS for automatic adaptation, which includes voice compression/duplication adjustment and admission control.

Empirical studies [24] showed that low-power wireless links are inherently lossy, which has a key impact on the quality of voice streams. For instance, our experiments on SenEar nodes with CC1100 radio showed that 5% packet loss can reduce the voice quality by up to 30%. Although several models have been proposed for quantifying voice quality, they are suitable for lossy wireless networks with ADPCM codec. We adopt an existing objective voice quality model called the E-model and measure its parameters for our platform using real voice samples.

Plenty of work has been done to model the quality of Voice over IP (VoIP). A widely adopted subjective method is called MOS (Mean Opinion Score) that qualifies speech quality with a score (in the range of 1 to 5) given by a group of audiences. A voice stream with MOS of 4.0 or higher is considered satisfactory, while a MOS of 2.6 or below is considered unacceptable. Although MOS has been shown effective in quantifying voice quality, it is a subjective measure and hence cannot be used for automatic voice evaluation and control in our system.

A widely used objective voice evaluation model is called E-model which was introduced by ITU in recommendation G.107 [3] in 1998. In the E-model, a quantity called R-value is derived from delays and equipment impairment factors. R-value is a single scalar ranging from 100 down to 0, which has the following relationship with MOS.

\[
MOS = \begin{cases} 
1; & R < 0 \\
4.5; & R > 100 \\
1 + 0.035R + 7 \cdot 10^{-6}R(R - 60)(100 - R); & \text{otherwise}
\end{cases}
\] (1)

From the above relationship, a MOS value of 2.6 corresponds to an R-value of 50, which is set as the threshold on acceptable voice quality. In [6], the R-value is given as

\[
R = R_0 - I_d - I_e
\]

where \( I_d \) and \( I_e \) are the quality degradation due to delay and codec-dependent equipment impairment factors, respectively. \( R_0 \) is the maximum R-value with no impairments, which is set to 93.2 corresponding to a MOS of 4.41. The \( I_d \) and \( I_e \) represent the voice quality loss due to delay and packet loss, respectively.

We now discuss the derivation of \( I_d \) and \( I_e \). The \( I_d \) is given by (3), where \( d \) is the end-to-end delay (in the unit of millisecond) composed of data compression delay, de-jitter buffer delay and network end-to-end delay. The \( I_d \)

\[
I_d = \alpha d + \beta x + \chi e
\] (3)

where \( \alpha \), \( \beta \) and \( \chi \) are three codec-dependent constants.

\[
R = R_0 - I_d - I_e
\] (2)

\[
I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3)
\] (3)

\[
I_e = \alpha + \beta \ln(1 + \chi e)
\] (4)

\[
H(x) = \begin{cases}
0; & \text{if } x < 0 \\
1; & \text{if } x \geq 0
\end{cases}
\]

\[
R = R_0 - I_d - I_e
\] (2)

\[
I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3)
\] (3)

\[
I_e = \alpha + \beta \ln(1 + \chi e)
\] (4)

\[
H(x) = \begin{cases}
0; & \text{if } x < 0 \\
1; & \text{if } x \geq 0
\end{cases}
\]

From (3), we can see that the impact of delay on voice quality is small. For instance, when the total delay is 177.3 ms, the decrease of \( R \) is only about 4.25. This is due to the use of buffer for voice playback at the receiver. This observation is consistent with our experimental results. In contrast, end-to-end packet loss plays a significantly more important role in voice quality. Therefore, we focus on obtaining the expression of \( I_e \) in (4).

The values of \( \alpha \), \( \beta \) and \( \chi \) in (4) have been obtained from VoIP experiments [6], which cannot be used for our system. We measured the values of \( \alpha \), \( \beta \) and \( \chi \) according to the method proposed in [20] using a set of 20 voice samples from ITU P.862 test set [18]. For each sample, the R-value is calculated from (1). The resulted pairs of packet loss rate and R-value are then used to obtain the values of \( \alpha \), \( \beta \) and \( \chi \) through curve fitting. The results are listed in Table 4. The fitted curves and measurements are shown in Fig 3.

<table>
<thead>
<tr>
<th>Codec Setting</th>
<th>( \alpha )</th>
<th>( \beta )</th>
<th>( \chi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADPCM-5bit</td>
<td>6.96</td>
<td>8.78</td>
<td>8.21</td>
</tr>
<tr>
<td>ADPCM-4bit</td>
<td>13.30</td>
<td>8.28</td>
<td>5.21</td>
</tr>
<tr>
<td>ADPCM-3bit</td>
<td>25.38</td>
<td>6.08</td>
<td>4.15</td>
</tr>
<tr>
<td>ADPCM-2bit</td>
<td>40.37</td>
<td>2.75</td>
<td>6.58</td>
</tr>
</tbody>
</table>

Table 1. The values of \( \alpha \), \( \beta \) and \( \chi \) in (4) measured under different ADPCM settings.

From (3), we can see that the impact of delay on voice quality is small. For instance, when the total delay is 177.3 ms, the decrease of \( R \) is only about 4.25. This is due to the use of buffer for voice playback at the receiver. This observation is consistent with our experimental results. In contrast, end-to-end packet loss plays a significantly more important role in voice quality. Therefore, we focus on obtaining the expression of \( I_e \) in (4).

The values of \( \alpha \), \( \beta \) and \( \chi \) in (4) have been obtained from VoIP experiments [6], which cannot be used for our system. We measured the values of \( \alpha \), \( \beta \) and \( \chi \) according to the method proposed in [20] using a set of 20 voice samples from ITU P.862 test set [18]. For each sample, the R-value is calculated from (1). The resulted pairs of packet loss rate and R-value are then used to obtain the values of \( \alpha \), \( \beta \) and \( \chi \) through curve fitting. The results are listed in Table 4. The fitted curves and measurements are shown in Fig 3.
5 Compression and Duplication Adaptation

As discussed in Section 4, the voice quality of a stream is severely affected by the packet loss on wireless links. Packet retransmission is a technique widely adopted to mitigate the impact of lossy links. However, it is not suitable for high-bandwidth voice streaming due to the high overhead and undesirable side-effects such as out of sequence packets in audio playback.

To deal with lossy wireless links, we adopt a simple reliability scheme that duplicates each data packet with certain probability, which has been shown effective for recovering audio packet loss [15]. There exists an interesting trade-off between audio compression ratio and duplication ratio. On the one hand, a higher duplication ratio reduces data loss by sending more redundant packets. On the other hand, it leads to more bandwidth usage, which causes the raw voice to be compressed at a higher ratio resulting in more quality loss. In this section, we present a mechanism for dynamically optimizing the compression and duplication ratios of a voice stream. This framework is used by QVS for achieving quality-assured voice streaming in face of dynamic wireless links.

Suppose the voice generated by source node \( i \) is \( A \) Kbps, and the compression ratio is \( c \). The source injection rate, denoted by \( \lambda(i) \), is defined as the raw data rate that source \( i \) sends to the network. Due to compression, \( \lambda(i) = A \cdot c \). Suppose a packet is duplicated at a probability of \( \gamma \in [0, 1] \). \( \lambda(i) \) can be expressed as:

\[
\lambda_{in}(i) = A \cdot c \cdot (1 + \gamma)
\]

Suppose the end-to-end packet loss rate is \( e \) and the packet loss events follow a Bernoulli process. We define data loss ratio (denoted by \( e' \)) as the probability that a data packet and its duplicate are lost. Data loss ratio can be derived from packet loss ratio and duplication ratio:

\[
e' = (1 - \gamma) \cdot e + \gamma \cdot e^2
\]

In (6), the \( e' \) is computed as the expected data loss in two cases where the packet has no duplicate (with the loss probability of \( e \)) and has a duplicate (with the loss probability of \( \gamma \cdot e^2 \)). According to (6) and (5), when the injection rate is fixed, a trade-off exists between compression ratio and duplication ratio. Intuitively, a higher duplication ratio mitigates the actual data loss while incurring more bandwidth usage. Consequently, the raw voice must be compressed at a higher ratio resulting in more quality loss. For known path loss ratio \( e \) and fixed injection rate \( \lambda(i) \), we can derive the combination of compression ratio \( c \) and duplication ratio \( \gamma \) that achieves the best voice quality (i.e., the minimum data loss \( e' \)).

We now use an example to illustrate the process of optimizing the compression and duplication ratios. When the injection rate is limited to 40 Kbps and the path loss ratio is 5%, we can compute the duplication ratio that minimizes the actual data loss ratio for each of the four ADPCM compression settings. From Table (5), we can see that the 4-bit ADPCM codec and a duplication ratio of 25% achieves the minimum quality loss.

### Table 2. The compression and duplication ratios for 40 Kbps injection rate and 5% path loss.

<table>
<thead>
<tr>
<th>Compression setting</th>
<th>Duplication ratio</th>
<th>Voice quality loss (( e' ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADPCM-5bit</td>
<td>0</td>
<td>32.987</td>
</tr>
<tr>
<td>ADPCM-4bit</td>
<td>25%</td>
<td>31.665</td>
</tr>
<tr>
<td>ADPCM-3bit</td>
<td>66.7%</td>
<td>31.669</td>
</tr>
<tr>
<td>ADPCM-2bit</td>
<td>100%</td>
<td>36.245</td>
</tr>
</tbody>
</table>

6 Stream Admission Control

We now present the stream admission control mechanism employed by QVS. Admission control determines whether a new voice stream can be accommodated in the network. Our admission control mechanism is based on rate allocation model that captures both the requirement of voice quality and the impact of interference on network capacity.

6.1 Saturation Rate and Local Capacity

In this section, we discuss a rate allocation model that is used in admission control of QVS. The purpose of rate allocation in our problem is to determine the data rate of each node such that the voice streams in the network can achieve satisfactory quality. This is different from existing work on wireless rate allocation whose primary objective is to ensure fair channel usage among different nodes or flows. We first briefly discuss several concepts used in our rate allocation model.

**Contention domain** of node \( i \) refers to the set of nodes that share the bandwidth with \( i \) (including \( i \) itself), i.e., whose transmissions interfere with \( i \). **Saturation rate** [4, 19] is the maximum throughput observed at a receiver when all senders are within the contention domain of...
each other. Fig. 4 shows the saturation rate versus the number of SenEar nodes that are within one hop of each other. The throughput reaches the maximum when there are two senders as the capacity of receiver is saturated. The throughput then drops gradually with more senders due to the increased overhead of channel contention.

Saturation rate of a contention domain depends on several factors such as spatial distribution of nodes and the design of MAC. Moreover, the number of nodes within a contention domain is difficult to obtain because the interference range is usually larger than the communication range. Although estimating contention domain and the associated saturation throughput in a multi-hop network is possible [17], it requires substantial overhead and complexity. In this work, we assume that the saturation rate is solely dependent on the number of nodes in a contention domain and hence it can be measured offline. We note that such assumption does not account for transient variation (e.g., due to environmental noise). However, QVS dynamically monitors each voice stream and controls the compression/duplication accordingly for robust voice quality. As a result, accurate knowledge of network capacity is not required. Moreover, we set the saturation rate conservatively for accounting for the variance of offline measurements. Suppose there exist $k$ nodes in a contention domain, we denote the saturation rate as $\lambda_{sat}(k)$.

In a multi-hop network, a node may belong to multiple contention domains. The local available capacity (or local capacity) of node $i$ is the maximum data rate it can send without causing the total data rate within any contention domain with which it is associated to exceed the saturation rate. We now discuss how to obtain the local capacity of node $i$, which will be used for rate allocation and admission control discussed later. Local capacity of each node is critical for admission control. We first define the following notation. $C(i)$ and $B_i$ represent the contention domain and local capacity of node $i$, respectively. $\lambda_{tot}(i)$ denote the total data rate of all voice streams that flow through node $i$.

The local capacity of a node is the minimum of all local available bandwidths of the nodes in its contention domain. This is because the data rate of $i$ contributes to the saturation throughput of all contention domains with which it is associated. Formally, node $i$ can calculate its local capacity by (7):$

\begin{equation}
B_i = \min_{j \in C(i)} \left( \lambda_{sat}(\|C(j)\|) - \sum_{k \in C(j)} \lambda_{tot}(k) \right)
\end{equation}

where $\|C(j)\|$ represents the cardinality of set $C(j)$.

### 6.2 Constraints of Rate Allocation

Our rate allocation model is mathematically described by a set of constraints related to channel capacity and voice quality. As we assume voice streams arrive one by one, we focus on the case of allocating data rate to a new voice stream.

Assume $s_1 \cdots s_{j-1}$ voice streams are being transferred in the network and stream $s_j$ is initiated. We denote $N_a(j)$ as the set of nodes that are actively transmitting at least one voice stream and their transmissions are affected by the new stream. Specifically, $N_a(j)$ consists of the nodes on the path from the source of $s_j$ and all the nodes which fall in the contention domains of these nodes. Suppose $N_p(j)$ represents the set of nodes that lie on the path from the source of $s_j$ to the sink. In order to admit a new stream into the network, two sets of constraints on stream quality and local capacity must be satisfied.

**Stream quality constraints:** If the new stream is admitted, the injection rates of all sources of the streams must be no lower than the threshold rate, e.g., enough to maintain the desired voice quality. The threshold rate of stream $s_i$, $\lambda_{th}(s_i)$, is derived from (2) and (4) using the minimum allowable R-value (50 in our design) and the packet loss ratio on the path of stream $s_i$.

\begin{equation}
\lambda_{in}(s_i) \geq \lambda_{th}(s_i), \quad \forall i \in [1, j]
\end{equation}

**Local capacity constraints:** For arbitrary node $i \in N_a$, the total rate of all streams flowing through itself and the nodes in its contention domain should be no higher than the local capacity $B_i$.

\begin{equation}
\|C(i) \cap N_p(j)\| \cdot \lambda_{in}(s_j) \leq B_i, \quad \forall i \in N_a(j)
\end{equation}

In (9), $\|C(i) \cap N_p(j)\|$ represents the number of nodes that lie in both the contention domain of $i$ and on the path of stream $s_j$. These nodes will share the local capacity with node $i$. Moreover, as all of them forward the data packets in stream $s_j$, the increased data rate of them is at most $\lambda_{in}(s_j)$.

![Figure 5. An 8-node network topology. The solid and dotted edges in the figure represent communication and interference links, respectively.](image)
the sink must satisfy. The solid and dotted edges in the figure represent communication and interference links, respectively. The set of nodes that are affected by the new stream is \( N_v(5) = \{1, 4, 5, 6, 7\} \). The constraints that must be met include:

\[
2\lambda_{in}(5) \leq B_5, \; 2\lambda_{in}(5) \leq B_6, \; 2\lambda_{in}(5) \leq B_4 \tag{10}
\]

\[
\lambda_{in}(5) \leq B_7, \; \lambda_{in}(5) \leq B_1 \tag{11}
\]

\[
\lambda_{in}(5) \geq \lambda_{th}(5) \tag{12}
\]

Constraints (10) to (11) are the local capacity constraints. For example, the transmissions of both node 5 and 4 will interfere with node 6 and hence the total data rate increase of them (2\(\lambda_{in}(5)\)) should not exceed the local capacity of node 6. Constraint (12) ensures that the voice quality meets the requirement.

6.3 Distributed Implementation

A new voice stream can be solicited by the sink or initiated by the source. We focus on the later case as the admission control for the two cases are similar. We first describe how local capacity is measured. Each node in the network maintains a stream table that contains the data rate of each stream in its contention domain. Each node periodically broadcasts its data rate (by piggybacking routing control messages). The local capacity can then be computed by each node according to (7).

The source that attempts to start a new stream first calculates its injection rate by (5). Note that the loss ratio of the path to the sink is required in (5). It can be calculated from the transmission count (ETX) of the path that is maintained by the routing layer [23]. The source then includes the calculated injection rate in an INIT message and sends to the sink and sets a timer \( T_{init} \). The sink sets a timer \( T_{ad} \) upon receiving the message. The nodes on the path from the source to the sink and those that overhear the message record the injection data rate. They then independently check if any of the constraints in (8) and (9) would be violated if the stream was admitted. Note that each node knows all the information required for this operation, which include the total data rate of streams in its contention domain, injection rate and local capacity. If a node sees a constraint violation, it notifies the sink. If the sink receives no notifications, it sends an GRANT message to the source and admits the new stream. Otherwise, it cancels the \( T_{ad} \) timer and eventually the \( T_{init} \) timer at the source fires, which indicates the failure of admission control. Note that the settings of \( T_{init} \) and \( T_{ad} \) must account for both the delay of network communication and the time cost of checking constraints of rate allocation. In particular, the duration of \( T_{ad} \) must be longer than that of \( T_{init} \) by a round-trip message delay.

6.4 Discussion

We now discuss several practical issues that are addressed in our design.

Multi-hop contention domain. It is well known that the interference range of a node is larger than the communication range. Therefore, the contention domain of a node may contain nodes that are not its immediate communication neighbor. Consequently, the total data rate of the streams in a node’s contention domain may be overestimated according to (7) which only considers the bandwidth usage of a node’s immediate neighbors. Although there exist techniques that [25] can measure multi-hop contention domains, they inevitably incur high overhead. To deal with this issue, we adopt a conservative approximation of the saturation throughput as discussed in Section 6.1, which counteracts the overestimated total data rate in the computation of local capacity.

Impact of route change and link quality variation. Several empirical studies [23] have shown that the low-power links on sensor nodes are subject to frequent quality variation. Moreover, network routes may migrate in response to such dynamics. As a result, the voice streams being transferred in the network may suffer from transient quality degradation. To deal with this issue, each source constantly measures the status of its stream according to voice quality model (2). When the average quality of a stream drops below a preset threshold, the source stops the transmission and then initiates the admission control process after a delay.

7 Experimentation

We implemented QVS on top of the routing layer of SenEar protocol stack (shown in Fig. 1). Our implementation has a code size of 4667 bytes and uses 236 bytes of RAM. We evaluate QVS using a 4-node simple chain network as well as a 20-node deployment in our lab illustrated in Fig. (9). The CC1100 radio on each node uses a transmit power of -20 dBm to obtain a multi-hop experimental network topology. To ensure satisfactory voice quality, we set the R-value threshold on the voice quality to be 50, as discussed in Section 4. The primary design goal of QVS is to transfer the maximum number of voice streams concurrently while the voice quality of each stream must meet the preset threshold.

7.1 Compression and Duplication Adaptation

We first evaluate the effectiveness of compression and duplication adaptation when the link quality varies. Four
nodes are placed on a straight line with the source and sink being the nodes at two ends. To create substantial variation in link quality, the sink is moved to three different locations that are further away from other nodes. The resulted average end-to-end packet loss ratio is 4.87%, 6.87% and 12.5%, respectively. When the source node detects the change of path quality (from the routing control messages propagated from the sink), it adjusts the compression and duplication ratios to maintain the required voice quality as discussed in Section 5.

We measure the R-value of the voice stream at the sink. The average R-value of voice stream in the three scenarios are 61.8, 57.4 and 59.2, respectively. The maximum standard deviation of the R-values is 5.4. This result shows that the dynamic compression and duplication adaptation is effective in maintaining good voice quality in presence of link quality variation. We now examine the system’s dynamic behavior in three scenarios.

Fig. 6 to 8 show the average packet loss ratio, the measured and expected data loss ratios. As discussed in Section 4, data loss ratio is the ratio of missing voice samples (whose packets and duplicate were lost during transmission). Based on the duplication ratio and measured packet loss ratio, the expected data loss ratio is computed according to (6) by assuming that the packet loss events are independent. When the packet loss ratio is 4.87%, Fig. 6 shows that both the measured and expected data loss ratios match the packet loss ratio well. This is because the duplication ratio is low (20%) and most data packets have only a single copy. When the packet loss ratio grows to 6.87% and 12.5%, the data loss ratio remains low (2.51% and 3.13%) as seen in Fig. 7 and Fig. 8.

Fig. 7 and Fig. 8 also show that the gap between the expected and measured data loss ratios increases. This is because more packets are lost in burst when links are lossier, which affects the error recovery from duplicate packets. Fig. 10 shows the histogram of consecutive packet loss events. We can see that most losses are due to single packet errors. Burst packet loss becomes evident only when the packet ratio increases to 12.5%. The overall results in Fig. 6 to 8 show that the dynamic compression and duplication adaptation can effectively maintain low data loss ratio and satisfactory voice quality.

We now evaluate the bandwidth usage of QVS with dynamic compression and duplication adaptation. For comparison, we used three fixed compression ratios supported by ADPCM that correspond to 32 Kbps, 24 Kbps and 16 Kbps data rates, respectively. We change the sink position of a 4-node chain network to create variation in end-to-end packet loss ratio. Fig. 11 shows the average voice quality when the average packet loss ratio is 8.4%. Due to the low loss ratio, QVS chooses to use a data rate of 24 Kbps with no duplication. We can see that the R-value of QVS stream remains above the specified bound of 50 and also is close to that of 32 Kbps stream, despite QVS uses 25% less bandwidth. In contrast, the R-value of 16 Kbps stream is below 50 in most of the time. Fig. 12 shows the average voice quality when the average packet loss ratio is 15.8%. To compensate the more packet loss, QVS chooses a 20% duplication ratio while keeping the compression ratio unchanged, which results in an actual data rate of 28.8 Kbps. The R-value of QVS stream remains
close to the specified bound of 50. In contrast, the 32 Kbps stream substantially overshoots the required voice quality while both 24 Kbps and 16 Kbps streams do not deliver satisfactory voice quality.

7.2 Admission Control and Capacity Utilization

We now examine the performance of admission control algorithm in QVS. Four source nodes, node 6, 8, 11 and 15 in the 20-node network (see Fig. 9) start to transmit a voice stream sequentially. The packet loss ratios of the paths from these nodes to the sink are 2.1%, 3.3%, 1.6% and 8.3%, respectively. The source nodes optimize the compression and duplication ratios accordingly, and select 16 Kbps for the first streams and 24 Kbps for the fourth stream. Fig. 13 shows the average R-value of all voice streams measured by the sink. We can see that the joining of a new stream causes the average voice quality of existing streams to drop temporarily. This is due to the transient contention on the channel despite the channel capacity has not been fully fulfilled. However, the voice quality quickly recovers from the drop and remains above the required threshold (50 R-value) in most of the time. When node 15 requested to start the fourth voice stream, nodes 2, 3, 4 and 5 found that their local capacity is not enough to accommodate the new stream. Thus the fourth stream would have failed the admission control. However, in order to test whether our admission control is conservative, we intentionally programmed all nodes to only record the admission control decisions while not notifying the sink. As a result, the fourth stream is allowed to join. Fig. 13 shows that the average R-value of all streams immediately drops below 50. Therefore, the admission control of QVS is able to admit the maximum number of streams in this experiment.

We now investigate the network capacity utilization of QVS. Fig. 14 shows the total data rate perceived by the sink and the total injection rate of all streams, which represent the goodput and offered load of the network, respectively. The two rates remain close to each other when there are no more than three voice streams as the network capacity has not been fully utilized. When the fourth voice stream joined the network, the offered load exceeds the goodput considerably. The goodput remains similar as before except a slightly higher variance is exhibited. This is because the network capacity has been used up by the first three streams and the fourth stream cannot improve the goodput further. Note that our admission control algorithm would have rejected the fourth stream if it was not intentionally disabled as described earlier. This result shows that the admission control of QVS leads to efficient network capacity utilization.

Several existing voice streaming protocols [15] have adopted fixed rate allocation schemes. We now compare QVS against such a scheme on the number of streams that can be supported. We implement a naive rate allocation scheme that chooses 32 Kbps, 24 Kbps or 16 Kbps (with no duplication) based on the path loss ratio such that the voice quality is above the threshold. Node 13, 5 and 10 shown in Fig. 9 sequentially start a new stream. The packet loss ratios of the paths from them to the sink are 5.2%, 1.7% and 3.1%, respectively. QVS selects 16 Kbps with 10% duplication ratio for the first stream. As the naive algorithm does not implement packet duplication, it must select a higher data rate 24 Kbps in order to to satisfy the required voice quality. Both algorithms use 16 Kbps with no duplication for the other two streams. We can see that the naive algorithm substantially overshoots the required voice quality resulting in bandwidth waste. Consequently, it only supports two concurrent streams with good voice quality. In contrast, the voice quality of QVS streams remains close to the required threshold and three voice streams can be transferred at the same time.

8 Conclusion

In this work, we designed and implemented a system for Quality-aware Voice Streaming (QVS) in WSNs. QVS is built upon a new sensor hardware platform for
high-rate audio communication. QVS consists of several novel components including automatic voice quality modeling and evaluation, dynamic voice compression and duplication adaptation, and distributed stream admission control. Our experimental results show that it delivers satisfactory voice quality under a range of realistic settings while achieving efficient network capacity utilization.

References